Real-time Speaker Independent Recognition of Bimodal Produced Speech

Boris Malčić, Vlado Delić, Jovan Galić and Nebojša Babić

***Abstract*—This paper presents the initial results in recognition of neutral speech and whispering in real-time, independent of a speaker. The speech database used for training is Whi-Spe. The system for training and testing is based on the Sphinx-4 recognition platform. The experiments in recognition showed average recognition accuracy of 86.2% (for normal speech) and 66.2% (for whisper). Compared to the recognition in controlled conditions, significant drop of the performance is observed in real-time recognition, for both speech modes.**

Keywords—Speech recognition; Whi-Spe speech database; Sphinx-4; whispered speech.

I. Introduction

Recent advances in automatic speech recognition (ASR) systems have brought many benefits in man-machine speech interaction. Despite good performance in controlled conditions, relatively high sensitivity in adverse conditions is the main reason for low robustness and poor use in real-life scenarios [1-2]. Speech technologies are intended for commonly used modes of speech, i.e. normally phonated speech (normal speech). Other speech modalities include shouted speech, louder speech, soft speech and whisper. The parameters for distinction of these 5 modes are sound pressure level (SPL), sentence duration and silence percentage, frame energy distribution and spectral tilt [3]. Whispered speech is pronounced as an alternative to neutral speech for a number of reasons: when someone does not like to disturb others, when loud speech is prohibited or unpleasant, when the information to speak is secret, when someone wishes to hide identity etc. Also, whispered speech can be produced due to health problems (laryngitis or rhinitis) [4]. Whisper as a speech mode is characterized by a lack of glottal vibration, noisy excitation of the vocal tract and in general, the changes of the vocal tract structure. Recent studies demonstrated performance gain in whispered speech recognition using data augmentation techniques [5-6], denoising autoencoders [7], as well as voice conversion [8].

The goal of the research study presented in this paper is to present initial results in real-time speaker independent recognition of bimodal produced speech (isolated words in neutral speech and whisper) for Serbian. The ASR system for training and testing is based on CMU Sphinx platform [9]. The remainder of this paper is organized in the following manner. In Section 2 a short overview of Hidden Markov Models (HMM) is given. Section 3 presents a speech database, the training of the ASR system and experiment setup for testing. In Section 4 we give results of experiments as well as its discussion, whereas concluding remarks and direction for future studies are stated in Section 5.

II. Hidden Markov Models

Modern ASR systems are inconceivable without Markov models that are combined with a model of a mixture of Gaussian distributions, or with deep neural networks that have become increasingly popular in recent times. The HMMs have become one of the most useful statistical methods for modeling speech signals [10]. Moreover, HMM is often defined by a parameter set $\mathbf{\Lambda} = (\mathbf{A}, \mathbf{B}, \boldsymbol{\pi})$ where are denoted: a transition probability matrix as $\mathbf{A} = [a_{ji}]$; an output probability matrix as $\mathbf{B} = [b_j(m)]$, $b_j(m) = P(X_t = o_m | x_t = j)$; and an initial state distribution matrix as $\boldsymbol{\pi} = [\pi_j]$. Furthermore, with a fully connected (ergodic) HMM, transition from any state to any other state is possible. On the other hand, due to the dynamic nature of the speech signal, in modern ASR systems, the serial structure dominates in which transition is possible only to a state with the index higher than the current one. An example of such structure is shown in Fig. 1 where the inactive states (null states) are shaded and located at the beginning and at the end of the sequence, while the output (ultimate) probabilities at the final moment $t = T$ (we observe a series of T feature vectors) are defined by $\eta_j = P(x_T = j), 1 \leq j \leq M, \eta_j + \sum_{i=1}^{M} a_{ji} = 1, \forall j$ for a certain state sequence $\boldsymbol{X} = \{x_1, x_2, \dots, x_T\}$.

In HMM-based speech recognizers, model parameters can be obtained using the Viterbi algorithm in training and the expectation maximization (EM) algorithm. Improved values are obtained by the EM algorithm using the method of Lagrange multipliers to determine the local extremum of a multivariate function. This procedure is called the iterative Baum-Welch (BW) training algorithm. Moreover, it is

Boris Malčić is with the Faculty of Electrical Engineering, University of Banja Luka, Department of Electronics and Telecommunications, Banja Luka, Patre 5, Bosnia and Herzegovina (e-mail: boris.malcic@etf.unibl.org), (https://orcid.org/0000-0002-7476-5140).

Vlado Delić is with the Faculty of Technical Sciences, University of Novi Sad, Novi Sad, Serbia (e-mail: vdelic@uns.ac.rs), (https://orcid.org/0000-0002-4558-9918).

Jovan Galić is with the Faculty of Electrical Engineering, University of Banja Luka, Department of Electronics and Telecommunications, Banja Luka, Patre 5, Bosnia and Herzegovina (e-mail: jovan.galic@etf.unibl.org), (https://orcid.org/0000-0002-2487-7136).

Nebojša Babić is with the Faculty of Electrical Engineering, University of Banja Luka, Department of Informatics, Banja Luka, Patre 5, Bosnia and Herzegovina (e-mail: nebojsa.baabic@gmail.com).

 ISBN 978-86-7466-930-3

important to emphasize that the BW algorithm does not guarantee reaching a global maximum but only a local one, and the iterative procedure is repeated while the joint probability of training data increases.

In order to be able to apply the previous discrete HMM methodology for speech signal recognition, continuously distributed HMMs have been introduced where the probabilities of emitting discrete symbols are changed as a function of the probability density of observations. Since the probability density function of any random variable can be approximated by the sum of *N* Gaussian random variables, the multivariate Gaussian mixtures are used in HMM recognizers.

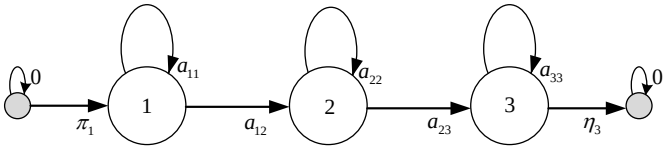

Fig. 1. An example of the serial left-right topology with 3 active states without the possibility of skipping a state

III. System for Recognition

A. Speech database

There are a relatively small number of languages with available speech databases created in modes other than neutral [11-13]. The Whi-Spe database was created for research activities needed for human-machine interaction in Serbian for bimodal produced speech [14]. In its initial form, the database contains recordings of 50 different words from 10 speakers (5 female and 5 male). The vocabulary of 50 words is divided in three groups: basic colors (6 words), numbers (14 words) and phonetically balanced words (30). Each word is repeated 10 times in both speech modes. Finally, the database includes 10.000 utterances (*wav* records) in a total duration of 2 hours. The sampling frequency of speech samples is 22050 Hz and 16 bits per sample (mono PCM *wav* format). More details about the database (recording, segmentation procedure, labeling and quality control) can be found in [14].

B. ASR training system

The task of the automatic speech recognition system (ASR) is to extract words from the speech signal in the order in which they are spoken. The block diagram of the ASR system based on HMM is shown in Fig. 2. Moreover, in section III c. we present a graphical interface of our program implemented in Java, and based on Fig. 2. First, in the feature extraction block, the speech signal is transformed into a series of feature vectors. The task of the feature extraction block is also to eliminate various variations caused by changes in speakers, ambience or channels. The task of the recognizer is to find the sequence of words that (according to a predetermined criterion) best corresponds to what was actually said.

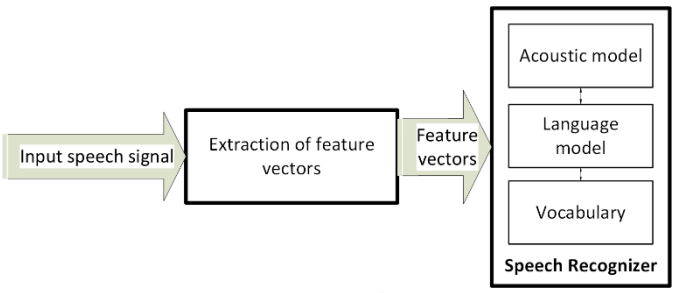

Fig. 2. The operation principle of a general ASR system

From a statistical point of view, the recognizer finds a string (i.e. $\widehat{W} = w_1, w_2, \dots, w_M$) of *M* words which maximizes a posteriori probability $P(W|X)$ where $\hat{X} = x_1, x_2, \dots, x_T$ is an array of *T* features. Namely, the language model describes the relationship between words taking into account the grammar of the language for which the recognizer is intended, while the description of the statistical point of view of speech behavior in the feature vector space is presented with an acoustic model. In ASR systems, the basic units for modeling are phonemes, and from their point of view in speech communications, the smallest acoustic unit that a person can perceive is a phoneme. In context-independent recognition, each phoneme is modeled independently, and that modeling unit is called a monophone. However, due to coarticulation, the pronunciation of a particular phoneme largely depends on neighboring phonemes, and then triphones (rarely biphones) are usually used, which are derived using the monophones in a way which takes into account the previous and next phonemes. When using the same, there is a need for a huge training database, which is also one of the main disadvantages of using triphones, and due to the relatively small Whi-Spe database used in this research paper, we only were able to train the acoustic model using the monophones.

Furthermore, let us now mention that speech recognition is divided into two phases: 1st phase of system training, 2nd phase of speech recognition or system testing. Common to both of these phases is that they need to perform feature vector extraction. In doing so, the feature vector should describe as adequately as possible the envelope of the amplitude spectrum of the spoken phoneme, because the information about the spoken phoneme is precisely contained in that envelope. In the training phase, we divided speech into frames (smaller segments) of 25 ms duration within which the speech signal can be considered quasi-stationary. Also, adjacent frames are shifted 10 ms to better track changes in the speech signal spectrum. The most commonly used features in speech recognition are the mel-frequency cepstral coefficients (MFCCs), and Fig. 3 presents the block diagram for the extraction of MFCC feature vectors.

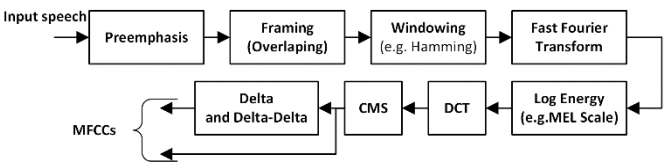

Fig. 3. Process for the MFCCs feature vector extraction

 ISBN 978-86-7466-930-3

Fig. 3. depicts that several steps need to be performed in the processing chain to extract MFCCs from the input speech signal. Some of these steps are Fast Fourier Transform (FFT) over speech signal frames. Next, the signal spectrum is passed through a filter bank (describing the operation of the basilar membrane) whose filters are distributed evenly on the mel-frequency scale. Furthermore, Discrete Cosine Transform (DCT) replaces inverse FFT and is used to calculate cepstral coefficients because the speech is a real signal and its amplitude spectrum is an even function. Then, to increase the robustness of the system, normalization with Cepstral Mean Subtraction (CMS) is performed, because by subtracting the mean value from Cepstral coefficients, a significant separation of excitation and transfer function of the vocal tract is achieved. This procedure is very useful in whisper recognition. Finally, with the last block in the processing chain on Fig. 3 we single out the dynamic features using which are better monitored the time characteristics of the speech signal, and that achieves less correlation between adjacent frames. For this purpose are used the so-called delta (Δ) features that represent the first derivative (rate of feature change) of static features, and also the delta-delta (ΔΔ) features that are obtained as a second derivative and represent the acceleration with which static features are changed.

In order to train the acoustic model for CMUSphinx, we followed in detail the procedure described in [15]. First, following the procedure all the necessary files (*an4.dic, an4.filler, an4.phone, an4_train.fileids, an4_train.transcription*) for training are created. For that purpose, the phonemes used in training the acoustic HMM for Serbian are presented in Table I. Next, in Table I phonemes {CC, CH, Dj, DZ, SH, ZH} are in Serbian {Ć, Č, Đ, Dž, Š, Ž}, and the notation Y is used for the phoneme SCHWA. Furthermore, an example of the phonetic transcription for only three words (of all 50 words in the Whi-Spe that are used in the process of training for the Serbian ASR system), is shown in Table II.

TABLE I
PHONEMES USED FOR MODEL TRAINING

A	B	C	CC	CH	D	Dj	DZ
E	F	G	H	I	J	K	L
Lj	M	N	Nj	O	P	R	S
SH	T	U	V	Y	Z	ZH	SIL

TABLE II
AN EXAMPLE OF PHONETIC TRANSCRIPTION FOR THREE WORDS

Word	Phonetic transcription
CRNA	C Y R Y N A
TRI	T Y R I
ZGRADE	Z G Y R A D E

Second, we prepared the directory **an4** (i.e. our appropriate directory where we created the *wav* subdirectory inside which are all *wav* records). In our case, 10,000 *wav* records are used from the training Whi-Spe database. After that, using the terminal, we positioned into the **an4**, and executed the command **sphinxtrain -t an4 setup**.

The previous command created a subdirectories **an4** and **feat** in the directory **an4** and the subdirectory **feat** was not visible in the file system, but it was visible in the terminal which could be checked with the commands **ls** and/or **la**. Please note that in relation to the steps of the official instructions, it is important that in the **etc** directory (created previously) from our database (which we called **an4**) we store all the necessary files for training, namely: *an4.dic, an4.filler, an4.phone, an4_train.fileids, an4_train.transcription, sphinx_train.cfg*. Next, be sure to create a wav directory in the directory of the database (**an4**), and place in it all wav records. Furthermore, we need to modify the configuration file (**sphinx_train.cfg**) by replacing the following lines (paths) in the configuration file:

```
$CFG_BASE_DIR = "/home/user/Desktop/sphinx-source/an4";
$CFG_SPHINXTRAIN_DIR = "/home/ user/Desktop/sphinx-source/sphinxtrain";
```

Also, in the configuration file we need to set

```
# (yes / no) Train contextually dependent models
$ CFG_CD_TRAIN = 'no';
```

Moreover, this previously modified **sphinx_train.cfg**, and all other files from the **etc** file within **an4**, should be copied and moved to the **etc** folder within the folder **/home/ user/Desktop/sphinx-source/sphinxtrain**. Finally, after all the previous preparation steps, we started the model training by calling **perl** scripts. Specifically, we go to the terminal in the **sphinxtrain** folder which is inside the **sphinx-source** directory (within that directory there are also **an4**, **sphinxbase** and **pocketsphinx** directories which we prepared earlier). Next, we observed that differences in github's **sphinxtrain** versions manifested by writing keyword **scripts** instead of the original **scripts_pl** in the training commands, and by changing the numbers of **perl** scripts equivalent to those in the training internals sections [15], so for example instead of the command **perl scripts_pl / 10.vector_quantize / slave.VQ.pl** now it is necessary to write **sudo perl scripts / 05.vector_quantize / slave.VQ.pl**. The previous is mentioned because we pulled the edge CMU Sphinx toolkit packages from github, and not used the recommended (5prealpha) releases [16]. Finally, after being positioned in the **sphinxtrain** directory, we executed the following commands in terminal for model training:

```
user@ ubuntu: ~ / Desktop / sphinx-source / sphinxtrain $ sudo perl scripts / 000.comp_feat / slave_feat.pl
user@ ubuntu: ~ / Desktop / sphinx-source / sphinxtrain $ sudo perl scripts / 00.verify / verify_all.pl
```

 ISBN 978-86-7466-930-3

user@ ubuntu: ~ / Desktop / sphinx-source / sphinxtrain $ sudo perl scripts / 05.vector_quantize / slave.VQ.pl
user@ ubuntu: ~ / Desktop / sphinx-source / sphinxtrain $ sudo perl scripts / 20.ci_hmm / slave_convg.pl.

After executing the last command, we got the parameters of the Context Independent (CI) trained model as in Fig. 4.

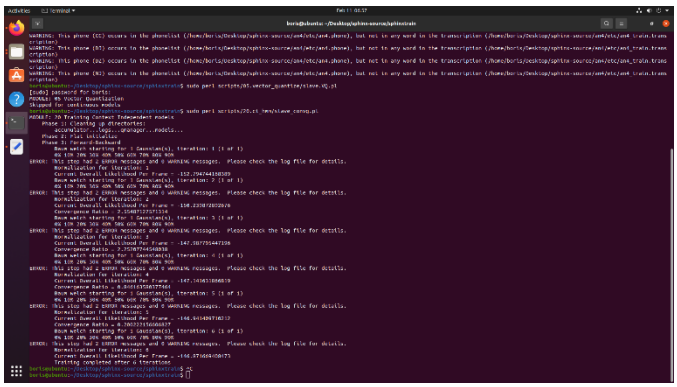

Fig. 4. An example of output **CI** training in the Ubuntu 20.04.3 LTS terminal

To sum up, we finally created new directories in **an4**, such as the **model_parameters** directory within which is located directory **an4.ci_cont** containing 7 files obtained during the training process, i.e.: **feat.params**, **mdef**, **mean**, **mix_weights, noisedict, transition_matrices**, **variances**. From the **feat.params** file is visible the number of filters in filterbank is 25. Then, as a feature vector, a 39-dimensional dynamic feature vector is used (13-static + Δ + ΔΔ). The number of Gaussian mixtures is 8. For each utterance, cepstral mean normalization (cmn) is performed. The number of monophones is 32 (of which 30 corresponding to the 30 phonemes in Serbian, then phoneme SCHWA and silence - SIL). As a result, the models of monophones are initialized with global mean value and variance (flat-start initialization). These all files in **an4.ci_cont** represent the CI continuous HMM that will be used in the following section by the application for the process of testing.

C. ASR recognition (testing) system

Fig. 5. shows the main window of the interface. Labeled with numbers, the components are as follows: 1. Main menu, 2. The list of words available in the model but left out, 3. The list of words chosen for the test, 4. Buttons for managing lists 2 and 3, 5. Most commonly used commands available in the main menu, 6. Output console, and 7. Console-related commands.

The main menu holds all options needed to run the test, load the language model, overview of the used grammar and management of extra words and sentences for the test. The application is packaged with the language model, but also any other compatible language model can be tested through the application. By default, the list of available words is displayed using the language model's dictionary.

When loading the custom model, the user can choose whether to use a grammar definition instead of relying on the language model.

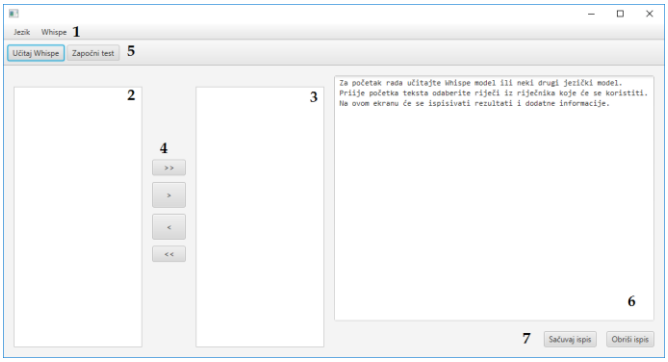

Fig. 5. The main window of the interface

As the list of available words loaded from the dictionary cannot match any but the most basic of grammars, the user can load sentences to use in the test. Each loaded sentence file is treated as a group of sentences where each sentence is one line in the text file. Each loaded file has the box (❒) icon next to the name to indicate that the selected keyword is actually a container of multiple test entries.

When the user wants to run a test, they have an option of shuffling and/or repeating the words multiple times. The user can also decide to save logs into a file instead of relying on the console for the test results. The console will show the results regardless of this setting. When performing many consecutive tests with the same settings, the user can instead provide the configuration through a file. When the file *testConfig.txt* is provided in the same directory as the application, the test settings step is skipped.

The test process screen is shown on Fig. 6. The test process aims to be as simple as possible. The left side of the window shows the instructions of what to pronounce. On each pronunciation, the next word or sentence is shown. The test can be stopped at any time for partial results. The right side of the window shows the test result logs. On each pronunciation, a line with a timestamp, the detected and the correct word will show. When the test is finished, the statistical result will also show, stating the overall accuracy. The console output can be saved or cleared at any time using the buttons below the console.

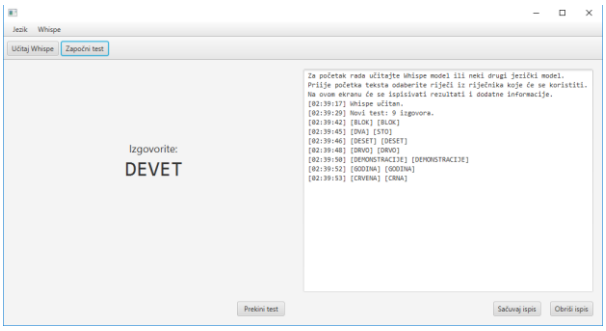

Fig. 6. The application window in the testing phase

 ISBN 978-86-7466-930-3

IV. RESULTS

The accuracy and correctness of speech recognition are tested on 10 people, i.e. 5 male testing examinees enumerated as {M1, M2, M3, M4, M5}, and 5 female examinees marked as {F1, F2, F3, F4, F5}. All the above respondents pronounced the words from Whi-Spe. The testing was performed in a room with SPL of 30-35 dB(A).

For this purpose, first the parameters of the HMM trained on the entire Whi-Spe for the normal speech, are loaded into the application, and then the testing scenario is repeated for the whisper speech. Exactly, these models are tested on all 10 examinees and this scenario is denoted as **full**. Then, we tested a new scenario denoted as **match** in which the parameters of HMMs were obtained by training only over part of the Whi-Spe (i.e. only on *female/male* speakers in the database), and the obtained models for normal speech (n) and whisper (w) are tested only on 5 *female/male* respondents, respectively. For all those models 8 Gaussian mixtures were used in the configuration file for the training. Next, for all different scenarios (*n_full*, *n_match*, *…*, *w_match*), the obtained corresponding results for the accuracy (***Acc*[%]**), and the correctness (***Corr*[%]**) are calculated in Table III and in Table IV, using the following expressions[17]

$$Corr[\%] = \frac{N-D-S}{N} \times 100\%, \quad (1)$$

$$Acc[\%] = \frac{N-D-S-I}{N} \times 100\%, \quad (2)$$

where: N is the total number of words in the reference transcriptions, D is the number of deletion errors, S is the number of substitution errors, and I is the number of insertion errors in each of the tested HMMs. From Table III it is calculated the mean value of recognition accuracy (*Acc_n_full*) of 86.4% with a mean absolute deviation (MAD) equal to 3.52% in the case of normal speech recognition on female examinees in the case of a full scenario (i.e. *n_full*), while for the whisper speech the mean *Acc_w_full* is 62.8% with the average correctness (*Corr_w_full*) of 90.8%. Furthermore, from Table IV, the mean of *Acc_n_match* is 88% (mean value of *Corr_n_match* is 93.6%) with MAD = 4% for testing the male examinees of normal speech for the **matched** (i.e. trained on utterances of male speakers, and tested on male examinees) case.

TABLE III
RESULTS OF SPEECH RECOGNITION ON FEMALE EXAMINEES FOR DIFFERENT SCENARIOS

Examinee	F1	F2	F3	F4	F5
Acc_n_full [%]	88	84	92	80	88
Acc_n_match [%]	74	88	74	70	62
Acc_w_full [%]	70	64	72	54	54
Acc_w_match [%]	50	58	66	56	60
Corr_n_full [%]	90	84	94	82	94
Corr_n_match [%]	86	92	88	84	92
Corr_w_full [%]	72	64	76	60	70
Corr_w_match [%]	58	72	74	68	78

TABLE IV
RESULTS OF SPEECH RECOGNITION ON MALE EXAMINEES FOR DIFFERENT SCENARIOS

Examinee	M1	M2	M3	M4	M5
Acc_n_full [%]	94	84	88	74	90
Acc_n_match [%]	84	86	84	94	92
Acc_w_full [%]	76	58	72	62	80
Acc_w_match [%]	76	76	82	62	80
Corr_n_full [%]	98	94	90	78	94
Corr_n_match [%]	94	94	88	96	96
Corr_w_full [%]	80	62	76	68	80
Corr_w_match [%]	80	84	86	64	88

Next, in the matched case for recognizing males' whispers the mean correctness is 80.4% with MAD = 6.72% and word error rate (WER) of 24.8%. Of course, in the case of the larger (with more different speakers) training database it would be logical to get better testing results in the match case, but for our research the relatively small Whi-Spe database was only available. Moreover, there is an evident reduction in real-time recognition accuracy compared to the controlled conditions (quiet environment and same recording equipment) in closed set speaker independent recognition based on HMM where accuracy was 98.3% (for neutral speech) and 96% (for whisper) [18]. As well, recognition of whisper in real-time is with significantly lower success than recognition of neutral speech. As can be seen from results in Tables 3 and 4, average recognition rate (accuracy and correctness) is higher for male speakers, but this is not statistically confirmed, despite the same training database. Variations among different speakers are high for both speech modes. For the determination of statistically significant parameters which contribute to high deviation of performance among speakers, a higher number of speakers is needed.

V. CONCLUSION

Speech recognition of mode other than neutral is by all means a serious challenge for modern ASR systems. In this paper, the experiments on real-time speech recognition in normal and whisper mode for Whi-Spe speech database and HMM algorithm, are conducted. Obtained results suggest that for recognition in real world scenarios, the larger speech database is needed for training. Future studies will be focused on the analysis of data augmentation techniques in multimodal speech recognition. Finally, the further research will also be based on deep neural networks (DNNs) because using *n*-gram language model from randomly initialized DNN with lattice-free maximum mutual information is possible WER relative reduction around 25% with respect to the best HMMs based ASR system [19].

ACKNOWLEDGEMENT

This research was supported by the Science Fund of the Republic of Serbia, #6524560, AI-S-ADAPT Ministry of

 ISBN 978-86-7466-930-3

Education, Science and Technological Development of Serbia: University of Novi Sad, Faculty of Technical Sciences (MPNTR - 451-03-68/2020-14/200156).

The authors would like to thank all examinees for participation in speech recognition tests.

REFERENCES

[1] M. P. Fernández-Gallego, D. Toledano, A Study of Data Augmentation for ASR Robustness in Low Bit Rate Contact Center Recordings Including Packet Losses, Applied Sciences, vol.2, no. 3, 1580, 2022.

[2] J. Holms, W. Holms, *Speech Synthesis and Recognition*, Taylor & Francis, London, United Kingdom, 2001.

[3] C. Zhang, J. H. L. Hansen, "Analysis and classification of Speech Mode: Whisper through Shouted," Proceedings of Interspeech 2007, pp. 2289-2292, 2007.

[4] T. Ito, K. Takeda, F. Itakura, "Analysis and recognition of whispered speech," Speech Communication, vol. 45, no. 2, pp. 139–152, 2005.

[5] P. R. Gudepu, G. P. Vadisetti, A. Niranjan, K. Saranu, R. Sarma, M. Ali Basha Shaik, and P. Paramasivam. "Whisper Augmented End-to-End/Hybrid Speech Recognition System - CycleGAN Approach." *INTERSPEECH* , 2020.

[6] H. Hikaru, and C. Pajot, "Data Augmentation for ASR using CycleGAN-VC.", Computer Science, 2021.

[7] Đ. Grozdić, S. Jovičić, M. Subotić, "Whispered speech recognition using deep denoising autoencoder", Engineering Applications of Artificial Intelligence, vol. 59, pp 15-22, 2017.

[8] M. Cotescu, T. Drugman, G. Huybrechts, J. Lorenzo-Trueba, A. Moinet, Voice Conversion for Whispered Speech Synthesis, IEEE Signal Processing Letters, 2020.

[9] P. Lamere, P. Kwok, E. Gouvêa, B. Raj, R. Singh, W. Walker, M. Warmuth, P. Wolf, "The CMU SPHINX-4 Speech Recognition System", ICASSSP, 2003.

[10] X. Huang, A. Acero, H-W. Hon, *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development*, 1st ed. New Jersey, USA, Prentice Hall PTR, 2001.

[11] P. X. Lee, D. Wee, H. S. Yin Toh, B. P. Lim, N. Chen, B. Ma, "Whispered Mandarin Corpus for Speech Technology Applications," Proceedings of Interspeech 2014, pp. 1598–1602, 2014.

[12] T. Tran, S. Mariooryad, C. Busso, "Audiovisual corpus to analyze whisper speech," presented at the International Conference on Acoustics, Speech and Signal Processing, pp. 8101–8105, 2013.

[13] F. Cummins, M. Grimaldi, T. Leonard, J. Simko, "The chains corpus: Characterizing individual speakers," Proceedings of International Conference on Speech and Computer SPECOM, St. Petersburg, Russia, pp. 421–435, 2006.

[14] B. Marković, S. T. Jovičić, J. Galić, Đ. Grozdić, "Whispered speech database: design, processing and application," In: Habernal, I., Matousek, V. (eds.), TSD 2013, LNAI 8082, Springer-Verlag Berlin Heidelberg, pp. 591–598, 2013.

[15] N. Shmyrev, 'Training an acoustic model for CMUSphinx'. [Online]. Available: https://cmusphinx.github.io/wiki/tutorialam/ . [Accessed: 11-Feb-2022].

[16] N. Shmyrev, 'CMU Sphinx downloads'. [Online]. Available: https://cmusphinx.github.io/wiki/download/ . [Accessed: 10-Feb-2022].

[17] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, P. Woodland, *The HTK Book (for HTK Version 3.4)*, Cambridge University Engineering Department, 2006. [Online]. Available: http://htk.eng.cam.ac.uk/prot-docs/htkbook.pdf .

[18] J. Galić, S. Jovičić, V. Delić, B. Marković, D. Šumarac Pavlović, Đ. Grozdić: "HMM-based Whisper Recognition Using μ-law Frequency Warping", SPIIRAS Proceedings, St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences, ISSN 2078-9181 (print), ISSN 2078-9599 (online), Issue No. 3(58), pp. 27-52, 2018.

[19] V. Delić, Z. Perić, M. Sečujski, N. Jakovljević, J. Nikolić, D. Mišković, N. Simić, S. Suzić, and T. Delić, "Speech Technology Progress Based on New Machine Learning Paradigm," *Comput Intell Neurosci.*, 2019. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6614991/. [Accessed: 05-Apr-2022].

 ISBN 978-86-7466-930-3